Article

Actuator Fault Detection and Fault-Tolerant Control for Hexacopter

Ngoc Phi Nguyen[ID]**, Nguyen Xuan Mung**[ID] **and Sung Kyung Hong** *

Department of Aerospace Engineering, Sejong University, Seoul 143-747, Korea;
phinguyen.183@gmail.com (N.P.N.); xuanmung1009@gmail.com (N.X.M.)
* Correspondence: skhong@sejong.ac.kr; Tel.: +82-02-3408-3772

check for updates

Abstract: In this paper, fault detection and fault-tolerant control strategies are proposed to handle the issues of both actuator faults and disturbances in a hexacopter. A dynamic model of a hexacopter is first derived to develop a model-based fault detection system. Secondly, the altitude control based on a sliding mode and disturbance observer is presented to tackle the disturbance issue. Then, a nonlinear Thau observer is applied to estimate the states of a hexacopter and to generate the residuals. Using a fault detection unit, the motor failure is isolated to address the one or two actuator faults. Finally, experimental results are tested on a DJI F550 hexacopter platform and Pixhawk2 flight controller to verify the effectiveness of the proposed approach. Unlike previous studies, this work can integrate fault detection and fault-tolerant control design as a single unit. Moreover, the developed fault detection and fault-tolerant control method can handle up to two actuator failures in presence of disturbances.

Keywords: fault diagnosis; hexacopter; fault-tolerant control; sliding mode control; Thau observer

1. Introduction

The multicopter unmanned aerial vehicles (UAVs) are drawing attention in the academic community. The have been developed and tested in several technologies, such as formation flight [1,2], precision landing [3,4], tracking control [5,6], remote sensing [7–10]. This significant growth has resulted from the following advantages that UAVs have, such as agility, economical cost, compactness, mechanical simplicity, and ability to operate in indoor and outdoor environments [11]. One of the control problems that a multicopter faces is actuator failure, which may cause many crashes and expose human beings to injury risks during operation [11]. Therefore, a fault-tolerant control (FTC) topic is a key factor in ensuring the safety and reliability demands of multicopters during flight missions. The goal of this paper is to propose a fault detection and fault-tolerant control method for hexacopters in the presence of one or two actuator failures.

1.1. Related Review

Studies on actuator fault detection (FD) and FTC for multicopter UAVs have been carried out intensively over the past decade. There are two types of failures in multicopter UAVs: partial loss and complete loss effectiveness in actuators. Several methods have been proposed to improve the stability and tracking performance for quadcopter UAVs in the presence of a partial actuator failure. Several control methods based on sliding mode control (SMC) [12–14] and adaptive SMC (ASMC) techniques [15–17] are proposed to handle model uncertainties, disturbances, and actuator faults. These studies show some acceptable results, but they cannot tolerate the fault in the case of complete failure [18,19]. Some control methods are investigated to overcome the complete failure issue. Merheb et al. suggested the reconfiguration technique based on control allocation to transform a quadcopter

into a tricopter [18]. The results can tolerate complete failure and demonstrate a good tracking performance. However, this method cannot detect the location of the fault and it requires an extra weight mounted on the opposite motor. There is another approach, based on backstepping [19], to transform the quadcopter into a birotor for emergency landing, but this method does not consider fault detection. In [20], a periodic solution combined with removing yaw motion was proposed to address one, two, or three propeller failures but this approach cannot use a fault detection scheme. Moreover, this work may not work in outdoor environments because GPS systems have a poor signal when removing yaw motion. It can be demonstrated that complete failure in one motor of a quadcopter requires removing yaw motion to ensure the controllability of the system because the quadcopter lacks redundancy in a dynamic model [11,20].

A hexacopter consists of six motors, including three motors rotating clockwise and three motors rotating counterclockwise. It was developed to increase the possibility of controllability of one or more actuator failures through its allocation matrix. Recent years have witnessed several works on the FTC problem of a hexacopter in the case of a motor failure. An attainable control set was proposed to evaluate the controllability of a hexacopter or an octocopter [21]. The optimal solution of thrust and torques is arrived at using parametric programming, which is obtained from the search tree method. In [22], an incremental backstepping fault-tolerant control is proposed for a hexacopter under an unknown control degradation occurring in the propulsion system. In [23], a Time Delay Control (TDC) method is presented to address single or multiple failure in the actuator. However, most of the above studies are tested and validated by simulation. A recent study presented experimental work on the FTC of a hexacopter using parametric programming formulation [24]. This work shows some good results in hovering or tracking performance after faults occur in one or two motors, but this design assumes that the location of motor failures is known to design the FTC approach. Although current studies proposed some good results on FTC of hexacopters, they lack experimental works or lack the fault detection unit to determine the location of a fault before designing the FTC method.

1.2. Main Contributions

In this paper, a fault-tolerant control approach is proposed for a hexacopter UAV to address one or two actuator faults with disturbances. Differing from previous studies, this article aims to integrate a fault detection unit into fault-tolerant control as a single unit. In detail, the fault detection unit is first used to detect the location of faults. Next, using this fault detection information, the motor failures are isolated and then a fault-tolerant control method is developed to hover the hexacopter through the reconfiguration technique.

This paper is organized as follows. Section 2 describes the dynamic model of the hexacopter. The attitude and altitude controller design are shown in Section 3. Section 4 presents the fault detection and isolation scheme. The experimental results are presented in Section 5 to validate the effectiveness of the proposed method. Section 6 presents the conclusions and the possibilities for future research.

2. Mathematical Model of a Hexacopter

Let us examine the body frame *B* and inertial frame *E* to describe a hexacopter dynamic (see Figure 1). For the body frame, the $X - Y$ plane is placed at the surface while the *Z*-axis is denoted by the right hand rule. The center of gravity (CoG) of the hexacopter is placed at the origin of the body frame. The body frame is transformed to inertia frame using rotation matrix in Equation (1).

$$R(\varphi, \theta, \psi) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi c\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - s\varphi c\psi \\ -s\theta & c\theta s\psi & c\theta c\varphi \end{bmatrix}, \tag{1}$$

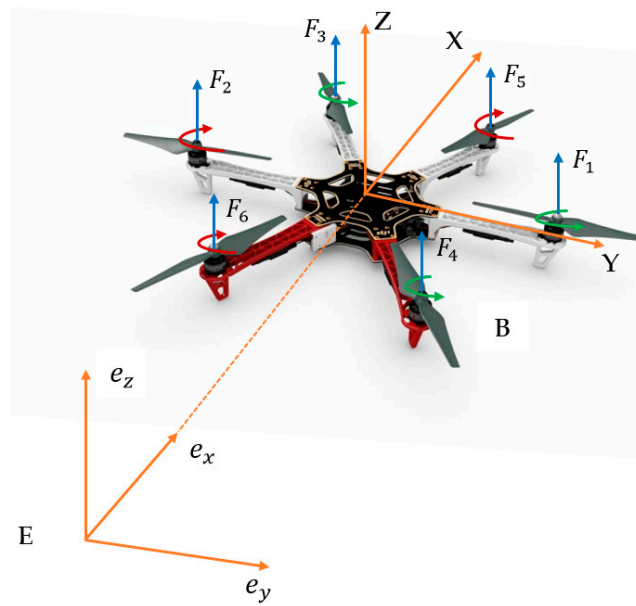where *s* denotes sin, *c* denotes cos, and φ, θ, ψ stand for Euler angles.

Figure 1. Hexacopter in the body and inertia frame.

The hexacopter shown in Figure 1 includes three motors (1, 3, 4) rotating counterclockwise and three motors rotating clockwise. Four control inputs U_1, U_2, U_3, and U_4 are defined as

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 \\ U_2 = (F_2 - F_1 + (F_3 + F_6 - F_4 - F_5)/2)L \\ U_3 = (F_3 + F_5 - F_4 - F_6)L\sqrt{3}/2 \\ U_4 = \tau_2 + \tau_5 + \tau_6 - \tau_1 - \tau_3 - \tau_4 \end{cases}, \tag{2}$$

where L is the arm length; $\tau_i = d\Omega_i^2$ and $F_i = b\Omega_i^2$ are the torques and forces generated from *ith* motor; Ω_i is the rotation speed; U_1 is the total thrust; U_2, U_3, U_4 are the torques in the directions φ, θ, ψ.

According to the Newton–Euler equation, the body dynamics is shown as [25]:

$$\begin{cases} m\ddot{r} = R\begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \dot{\omega} \times m\ddot{r} \\ I\ddot{\omega} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} - \dot{\omega} \times I\ddot{\omega} \end{cases}, \tag{3}$$

where m is the mass of the hexacopter; ω is the angular velocity vector; g is the gravity; I is the inertia vector; r is the position in inertial frame;

Finally, the orientation and translation motion of the hexacopter can be derived as [26]

$$\begin{cases} \ddot{x} = \{U_1(\cos\varphi sin\theta \cos\psi + \sin\varphi \sin\psi) - K_x\dot{x}\}/m \\ \ddot{y} = \{U_1(\cos\varphi sin\theta sin\psi - \sin\varphi \sin\psi) - K_y\dot{y}\}/m \\ \ddot{z} = -g + \{U_1(\cos\varphi cos\theta) - K_z\dot{z}\}/m \\ \ddot{\varphi} = \left(U_2 + (I_y - I_z)\dot{\theta}\dot{\psi} - J_T\dot{\theta}\Omega - K_\varphi\dot{\varphi}\right)/I_x \\ \ddot{\theta} = \left(U_3 + (I_z - I_x)\dot{\varphi}\dot{\psi} - J_T\dot{\varphi}\Omega - K_\theta\dot{\theta}\right)/I_y \\ \ddot{\psi} = \left(U_4 + (I_x - I_y)\dot{\varphi}\dot{\theta} - K_\psi\dot{\psi}\right)/I_z \end{cases}, \tag{4}$$

where Ω is the disturbance: $\Omega = \Omega_3 + \Omega_4 - \Omega_1 - \Omega_2$; U_1 is the total thrust; U_2, U_3, U_4 are the torques in the directions φ, θ, ψ; I_x, I_y, I_z are the moments of inertia of the hexacopter along the x, y, z axis; K_φ, K_θ, K_ψ, K_x, K_y represent drag coefficients; J_T is the inertia of each rotor.

From Equation (4), the relationship between control variables and forces can be described by the allocation matrix T as:

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -L & L & L/2 & -L/2 & -L/2 & L/2 \\ 0 & 0 & L\sqrt{3}/2 & -L\sqrt{3}/2 & L\sqrt{3}/2 & -L\sqrt{3}/2 \\ -d/b & d/b & -d/b & -d/b & d/b & d/b \end{bmatrix}. \tag{5}$$

3. Attitude and Altitude Controller Design

3.1. Attitude Controller Design

The full control structure is shown in Figure 2. In this scheme, the disturbance-based sliding mode control method is proposed for attitude control and to handle disturbances, whereas the fault detection and isolation unit is used to detect fault occurrence. When the residuals from the fault detection unit exceed the threshold, the motor fault is isolated and then the allocation matrix is modified correspondingly to adopt the new geometry of the hexacopter.
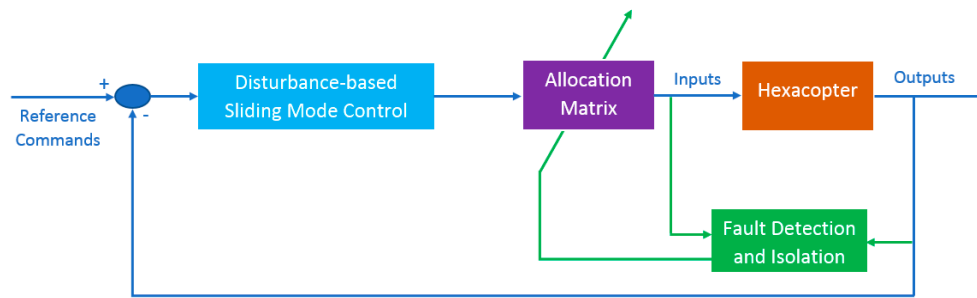

Figure 2. Control structure of a hexacopter.

Define $x = \begin{bmatrix} \varphi & \dot{\varphi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}^T$ and $U = \begin{bmatrix} U_2 & U_3 & U_4 \end{bmatrix}^T = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$ as the state vector and control input vector, respectively. From Equation (4), each rotational movement equation has the following form:

$$\begin{cases} \dot{x}_{2i-1} = x_{2i} \\ \dot{x}_{2i} = f_i(x) + h_i u_i(t) + d_i(x, t) \end{cases} \tag{6}$$

The desired attitude is denoted as x_i^d. The control goal is to generate the control signal u_i such that the hexacopter can track the desired attitude, i.e., $x_{2i-1}(t) \to x_i^d$ as $t \to \infty$, $i = 1, 2, 3$.

Assumption 1. *The disturbance is assumed to be bounded and satisfied $|d_i| \leq D_i$.*

Assumption 2. *The disturbance changes slowly and satisfies $\dot{d}_i = 0$.*

Assumption 3. *The estimation error of disturbance is bounded by $\left| \dot{\tilde{d}}_i \right| \leq \Gamma_i$.*

According to [27], a nonlinear disturbance-based sliding mode control is proposed by

$$\begin{cases} \hat{d}_i = z_i + \delta_i x_{2i} \\ \dot{z}_i = -\delta_i z_i - \delta_i(\delta_i x_{2i} + f_i(x) + g_i(x)u_i) \end{cases}, \tag{7}$$

where $\hat{d}_i$ is the estimate of d_i, z_i is the observation auxiliary vector, $\delta_i > 0$ is the observer gain. The estimation error of disturbance is denoted as:

$$\widetilde{d}_i = d_i - \hat{d}_i. \tag{8}$$

From Assumption 2, the derivative of disturbance estimation error can be defined as:

$$
\begin{aligned}
\dot{\widetilde{d}}_i &= \dot{d}_i - \dot{\hat{d}}_i \\
&= -\dot{z}_i - \delta_i \dot{x}_{2i} \\
&= \delta_i(\hat{d}_i - \delta_i x_{2i}) + \delta_i(\delta_i x_{2i} + f_i(x) + g_i(x)u_i) - \delta_i(f_i(x) + g_i(x)u_i + d_i) \\
&= -\delta_i \widetilde{d}_i
\end{aligned} \tag{9}
$$

Denote the control error:

$$e_i = x_{2i-1} - x_i^d . \tag{10}$$

The sliding surface is expressed by:

$$s_i = \dot{e}_i + k_i e_i , \tag{11}$$

where k_i is the positive parameter.

Taking the derivative of sliding surface

$$\dot{s}_i = \ddot{e}_i + k_i \dot{e}_i . \tag{12}$$

From Equations (7)–(9), $\dot{s}_i$ can be derived as

$$\dot{s}_i = f_i(x) + h_i u_i(t) + d_i(x,t) - \ddot{x}_i^d + c_i \dot{e}_i . \tag{13}$$

Theorem 1. *Consider the sliding surface is designed by Equation (11). Suppose that the following control law is implemented by*

$$u_i = \left(-f_i(x,t) - \hat{d}_i(x,t) + \ddot{x}_i^d - c_i \dot{e}_i - \Gamma_i \text{sign}(s_i)\right)/h_i . \tag{14}$$

Then the nonlinear system (Equation (6)) is stable and the control errors are forced to zero.

Proof of Theorem 1. Choose the Lyapunov function as follows [26]

$$V = \frac{1}{2}s_i^2 + \frac{1}{2}\widetilde{d}_i^2 . \tag{15}$$

Take the first derivative of the Lyapunov function

$$
\begin{aligned}
\dot{V} &= s_i \dot{s}_i - \delta_i \widetilde{d}_i^2 \\
&= s_i(\widetilde{d}_i - \Gamma_i \text{sign}(s_i)) - \delta_i \widetilde{d}_i^2 \\
&\leq -\left(\Gamma_i - \left|\widetilde{d}_i\right|\right)|s_i| - \delta_i \widetilde{d}_i^2 \\
&\leq 0
\end{aligned} \tag{16}
$$

Therefore, according to the control strategy (Equation (14)) and disturbance observer (Equation (9)), the closed-loop system can maintain stability when both actuator faults and external disturbances occur. $\square$

Remark 1. *With the proposed disturbance observer-based sliding mode control, the discontinuous control gain is decreased, and it is required to be larger than the bound of estimation error rather than that of disturbance.*

Remark 2. *To handle the chattering issue due to the sign function from Equation (14), a saturation function is denoted as*

$$sat(s_i) = \begin{cases} s_i & \text{if } |s_i| \leq 1 \\ sign(s_i) & \text{if } |s_i| > 1 \end{cases}, \quad i = 1, 2, 3. \tag{17}$$

3.2. Altitude Controller Design

A PID controller is applied to control the vertical movement of the hexacopter as follows

$$\ddot{Z} = \ddot{Z}_d + K_{dz}(\dot{Z} - \dot{Z}_d) + K_{pz}(Z - Z_d) + K_{iz} \int (Z - Z_d)dt, \tag{18}$$

where Z_d are the desired altitude in the z directions, respectively; Z is actual value; and K_{pz}, K_{dz}, and K_{iz} are the controller gains.

4. Fault Detection and Fault-Tolerant Control System

Fault detection consists of two modules: residual generation and residual validation. Residual generation is defined as the difference between system output and state observer. The information of residual generation is used for the residual validation module that determines fault-free or faulty case. When the output of residual validation module is in the faulty case, the fault isolation (FI) module is activated. The obtained FI information is used to determine the location of the fault, which is based on the magnitude and sign of residual validation module. Finally, the fault-tolerant control system aims to change the allocation matrix to hover and land the hexacopter.

4.1. Residual Generation

A nonlinear observer presented in [28] is applied for fault detection and isolation (FDI). Let us consider the following nonlinear model:

$$\begin{cases} \dot{x}(t) = Ax(t) + BL_1u(t) + h(x(t), L_1u(t)) \\ y(t) = Cx(t) \end{cases}, \tag{19}$$

where $x(t) = \begin{bmatrix} \varphi & \dot{\varphi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} \end{bmatrix}$, $u(t) = \begin{bmatrix} U_2 & U_3 & U_4 & U_1 \end{bmatrix}$, $y(t)$ are the state vector, control input vector, and the output vector; A, B, C are the system matrices; L_1 is the fault matrix; $h(x(t), L_1u(t))$ is the nonlinear function.

From the state model in Equation (19), the following conditions are designed for a Thau observer [28]:

Condition 1. *the pair (C, A) is observable.*

Condition 2. *the nonlinear part $h(x(t), u(t))$ is Lipschitz and differentiable continuously with a constant γ, i.e., $\|h(x_1(t), u(t)) - h(x_2(t), u(t))\| \leq \gamma\|x_1 - x_2\|$.*

If all conditions are satisfied, the Thau observer can be derived as [28]:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + K(\hat{y}(t) - y(t)) + h(\hat{x}(t), u(t)) \\ \hat{y}(t) = C\hat{x}(t) \end{cases}, \tag{20}$$

where $\hat{x}(t) = \begin{bmatrix} \hat{\phi} & \dot{\hat{\phi}} & \hat{\theta} & \dot{\hat{\theta}} & \hat{\psi} & \dot{\hat{\psi}} & \hat{z} & \dot{\hat{z}} \end{bmatrix}$ is the state observer vector; $\hat{y}(t)$ is the observer output vector; K is the designed matrix and determined by

$$K = P_{\varepsilon}^{-1}C^{T}. \tag{21}$$

The matrix P_{ε} is achieved from the following equation

$$A^{T}P_{\varepsilon} + P_{\varepsilon}A - C^{T}C + \varepsilon C^{T}P_{\varepsilon} = 0, \tag{22}$$

where ε is a positive value such that $P_{\varepsilon} \geq 0$.

The residuals are defined as the deviation between real output of system and state output:

$$\begin{aligned} r_1 &= \varphi - \hat{\varphi}, \ r_2 = \theta - \hat{\theta} \\ r_3 &= \psi - \hat{\psi}, \ r_4 = z - \hat{z} \\ r_5 &= \dot{\varphi} - \dot{\hat{\varphi}}, \ r_6 = \dot{\theta} - \dot{\hat{\theta}} \\ r_7 &= \dot{\psi} - \dot{\hat{\psi}}, \ r_8 = \dot{z} - \dot{\hat{z}} \end{aligned} \tag{23}$$

4.2. Residual Validation

The residual validation module is used to detect the fault-free or faulty case of system, which is compared with upper or lower thresholds. The decision logic is determined as follows:

$$\begin{cases} r_i(t) \geq H_i \text{ or } r_i \leq M_i \text{ in fault-free case} \\ r_i(t) < H_i \text{ and } r_i(t) > M_i \text{ in faulty case} \end{cases}$$

.

Using this residual validation module, the fault isolation and fault-tolerant control methods will be designed in Section 4.3 to isolate the fault and to reconfigure the allocation matrix.

4.3. Fault Isolation and Fault-Tolerant Control

When residual validation of FD system detects the faulty case, the FI module is activated. To determine the location of the fault, the knowledge of residuals r_5, r_6, r_7, r_8 should be used. The failure motor is isolated through the sign and magnitude of these residuals. The map of failure in one or two motors can be concluded in Tables 1 and 2, which depends on the sign and magnitude of residuals. In these Tables, the "Controllable" column means that the hexacopter can hover its position and make a landing mode during flight test. After the fault is isolated from the FI module, the allocation matrix is reconfigured correspondingly for fault accommodation.

Table 1. Map of one motor failure.

Motor Failure	r_5	r_6	r_7	r_8	Controllable
f_1	+	0	+	−	Yes
f_2	−	0	−	−	Yes
f_3	−	−	+	−	No
f_4	+	+	+	−	Yes
f_5	+	−	−	−	No
f_6	−	+	−	−	Yes

Where f_i is the fault occurring on motor i; the sign "−" denotes a normal negative value; the sign "+" denotes a normal positive value.

Table 2. Map of two motor failures.

Motor Failure	r_5	r_6	r_7	r_8	Controllable
f_{12}	0	0	0	−−	Yes
f_{13}	+	−	++	−−	No
f_{14}	+	+	++	−−	No
f_{15}	+	−	−	−−	No
f_{16}	+	+	−	−−	Yes
f_{23}	−−−	−	+	−−	No
f_{24}	−	+	+	−−	Yes
f_{25}	−	−	−−	−−	No
f_{26}	−−	+	−−	−−	No
f_{34}	−	−	++	−−	No
f_{35}	0	−	0	−−	No
f_{36}	−−	−	+	−−	No
f_{45}	+	−	−	−−	No
f_{46}	0	+	0	−−	No
f_{56}	+	+	−−	−−	No

Where f_{ij} is the fault occurring on motor i and j, $i < j$; the signs "−−", "−−−" denote "small negative value", and "very small negative value"; the signs "++", "+++" denote "big positive value", and "very big positive value".

5. Experimental Results

5.1. Experimental Setup

The fault-tolerant control, fault detection, and fault isolation methods in Sections 3 and 4 were tested on a DJI F550 platform [29]. These algorithms were developed on Pixhawk2 flight controller [30], which requires the C++ program for implementation [31]. The firmware version 3.5 was chosen for the flight controller. During testing, the faults were injected by limiting the pulse width modulation (PWM) [32] through a remote control that can switch to faulty mode from stabilize mode. The flight data were monitored using Mission Planner (MP) [33] software through Xbee wireless communication [34]. The fault detection data were achieved through a log file that is coded in C++ by the user. The experimental procedure is summarized in Figure 3.
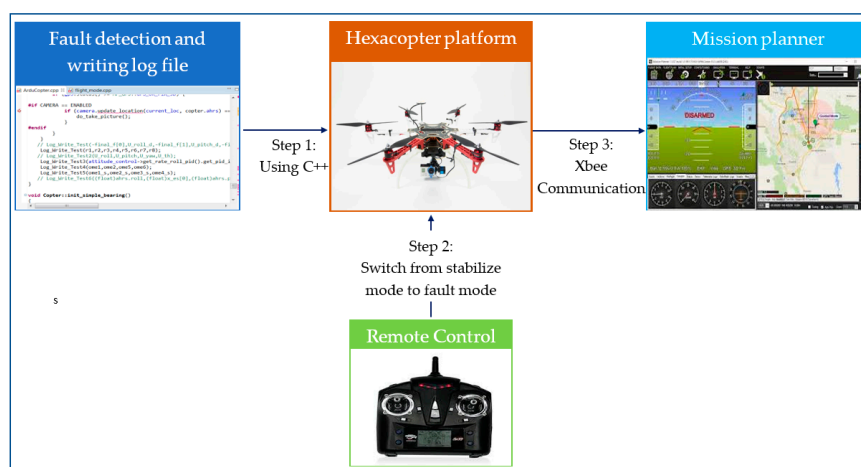
Figure 3. Procedure of the experimental test.

5.2. Results

To demonstrate the performance of the proposed control scheme, different simulation tests on a hexacopter are considered. The hexacopter parameters are shown in Table 3.

Table 3. Hexacopter parameters.

Parameter	Description	Value
L	Arm length	0.275 m
b	Thrust coefficient	1.177×10^{-5} N/m^2
d	Drag coefficient	1.855×10^{-7}
m	Mass	2.5 kg
$I_x; I_y; I_z$	Moments of inertia	0.00915; 0.00915; 0.01187 kg·m^2

The parameters of disturbance-based sliding mode control are chosen as: $k_i = 10$, $\delta_i = 3$, $i = 1, 2, 3$. The parameters of fault detection scheme are chosen as: $M_i = -0.35$ and $H_i = 0.35$. $A =$

$$\begin{bmatrix} 0_{4\times4} & I_{4\times4} \\ 0_{4\times4} & 0_{4\times4} \end{bmatrix}, \quad K = \begin{bmatrix} 1.1 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 \\ 0 & 0 & 1.1 & 0 \\ 0 & 0 & 0 & 1.1 \\ 0.3025 & 0 & 0 & 0 \\ 0 & 0.3025 & 0 & 0 \\ 0 & 0 & 0.3025 & 0 \\ 0 & 0 & 0 & 0.3025 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$h(x,u) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ (\dot{\theta}\dot{\psi}(I_y - I_z) - J_T\dot{\theta}\Omega)/I_x \\ (\dot{\phi}\dot{\psi}(I_z - I_x) - J_T\dot{\phi}\Omega)/I_y \\ \dot{\phi}\dot{\theta}(I_x - I_y)/I_z \\ -g \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -L/I_x & 0 & L/I_x \\ -L/I_y & 0 & L/I_y & 0 \\ d/(bI_z) & -d/(bI_z) & d/(bI_z) & -d/(bI_z) \\ 1/m & 1/m & 1/m & 1/m \end{bmatrix}.$$

In scenario 1, partial loss effectiveness in actuator 2 is presented to verify the disturbance-based sliding mode control and residuals from Table 1. To show the fault detection and isolation effectiveness, scenarios 2, 3, and 4 show the case of one motor failure, two motor failure in sequence, or two motor failure simultaneously.

5.2.1. Partial Loss in Actuator 2

A 30% loss of control effectiveness in the actuator fault 2 is simulated. The hexacopter can hover at the altitude of 2 m and the fault occurs at t = 28 s. It can be seen in Figure 4 that the Euler angles can converge quickly to the desired angles after a fault occurrence due to the disturbance observer-based controller. Moreover, the actual altitude can also track the desired one after the fault occurrence.
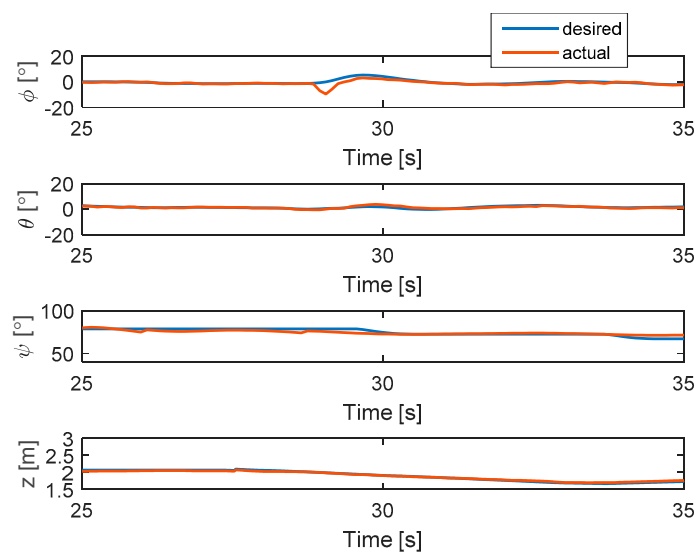
Figure 4. States of the system.

The residuals are shown in Figure 5. Interestingly, the residuals are the same as those in the fault map presented in Table 1. Since the residuals do not exceed the thresholds, the attitude still uses the disturbance-based sliding mode control for accommodation without changing the allocation matrix.
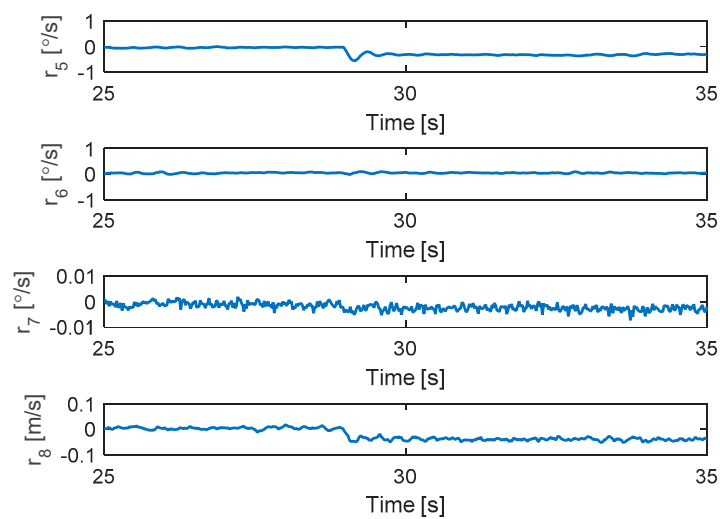

Figure 5. Residuals generated from fault detection schemes.

The output signals are illustrated in Figure 6. It should be noted that before the fault (up to t = 28 s), the six PWM signals are similar. After fault is injected in motor 2, the second PWM is decreased and then its value increases again to maintain the position of the hexacopter.
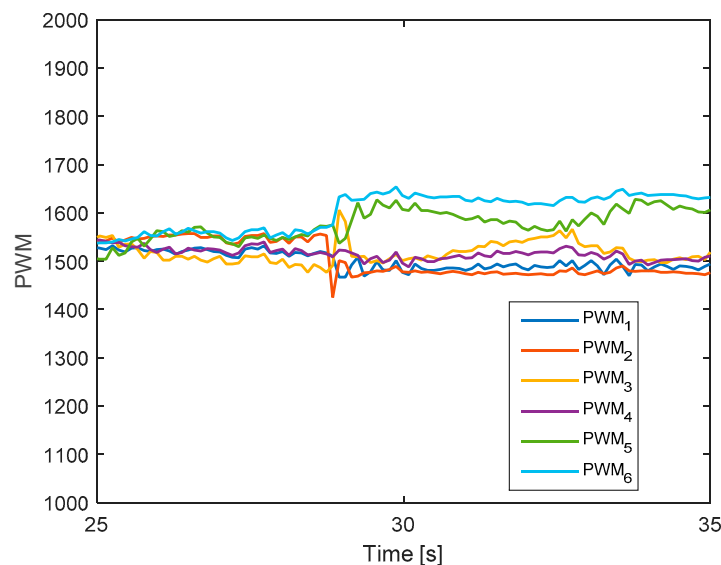
Figure 6. Output signals of the system.

The experimental work can be found at https://www.youtube.com/watch?v=en-h_HwVh0k or in the Supplementary Materials.

5.2.2. Complete Loss in Actuator 4

In this test, complete failure is introduced to motor 4. The hexacopter is made to hover at the height of 2 m and the fault is injected at t = 47.8 s. As can be seen in Figure 7, the attitude angles can track the desired angles although the they have some small oscillations after a fault occurs. The residuals of system are presented in Figure 8. It is shown that when fault occurs, all signs of residuals are similar to those in Table 1 at the beginning and then they go back to the zero. Unlike previous case, the residuals here converge to the original point because the magnitude of residuals exceed the threshold and then the motor 4 is isolated after fault occurrence.
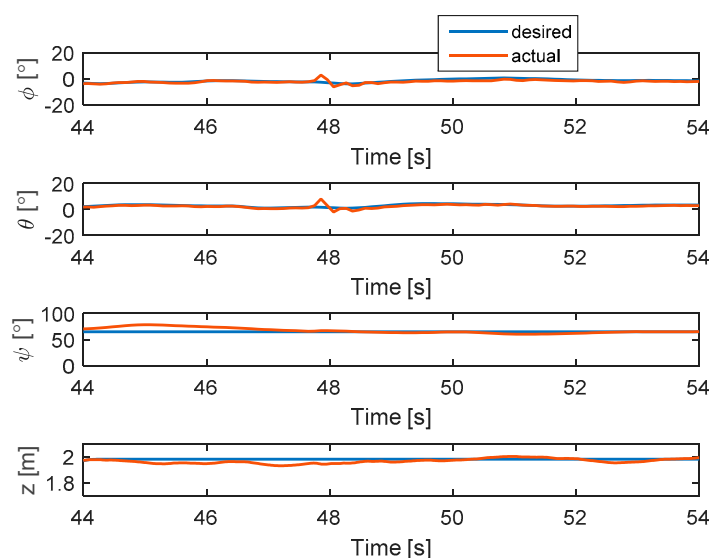
Figure 7. States of the system.
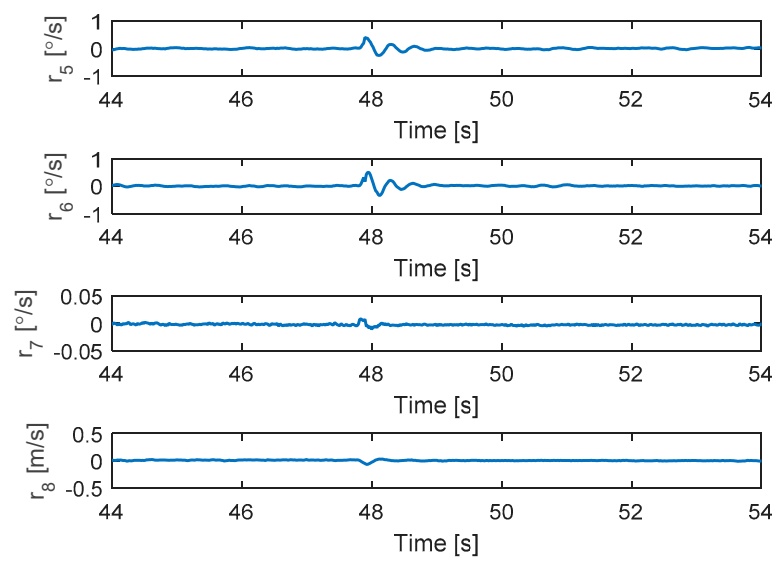
Figure 8. Residuals generated from fault detection schemes.

The detection unit of motors 2 and 4 is presented in Figure 9. It is clear that only motor 4 is detected from the FD unit. Figure 10 shows the corresponding output signals. It shows that the PWM of motor 4 has a minimum value due to fault injection. The other PWMs are changed correspondingly to maintain the position of hexacopter because the motor 4 is isolated from the allocation matrix.
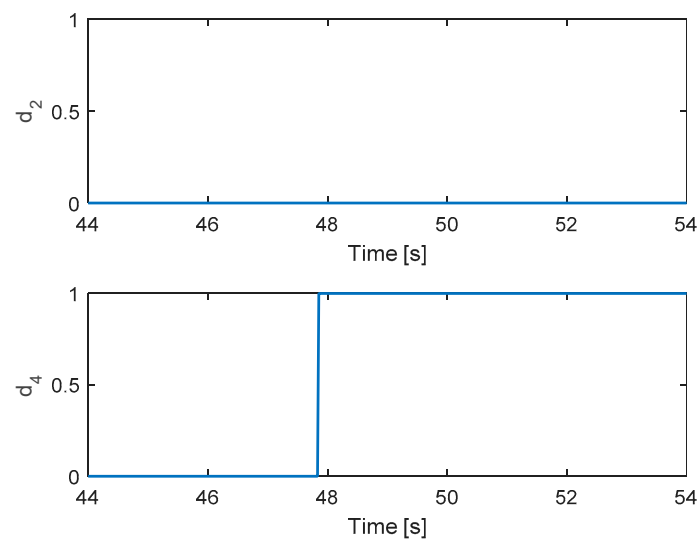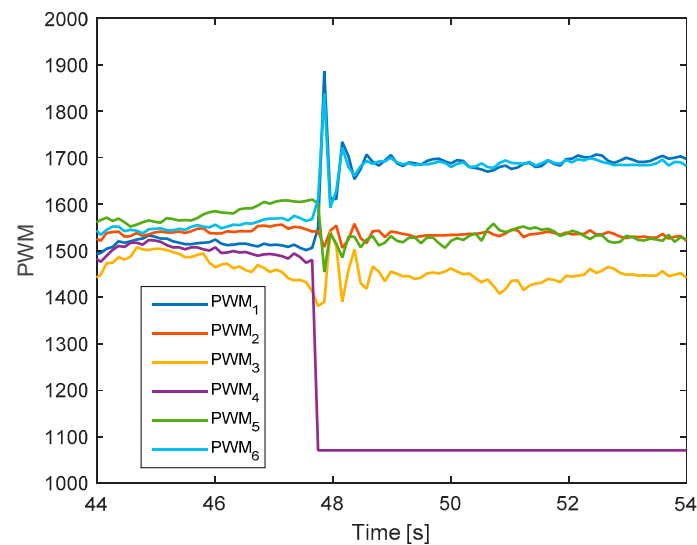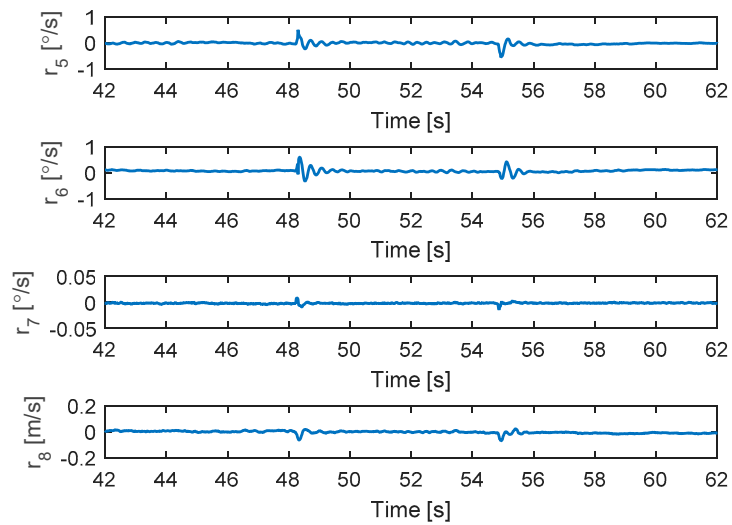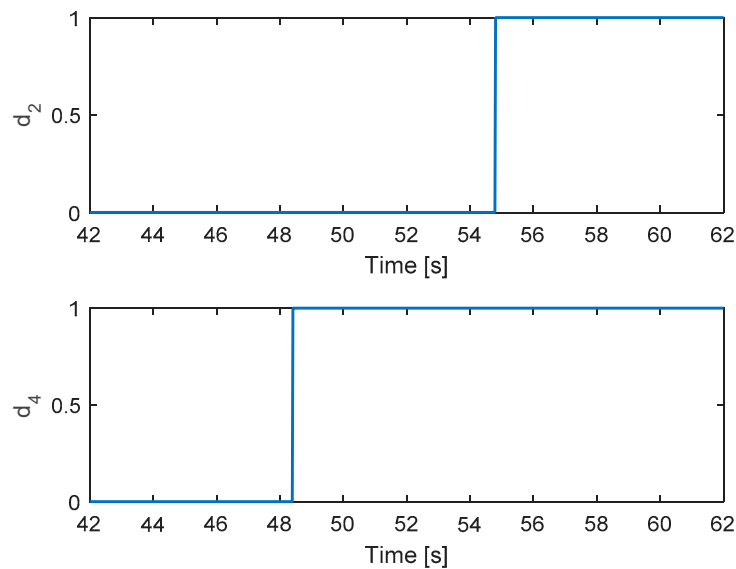

Figure 9. Fault detection of actuators 2 and 4.

Figure 10. Output signals of the system.

The experimental work can be found at https://www.youtube.com/watch?v=rSSoVS4nDaE or in the Supplementary Materials.

5.2.3. Complete Loss of Two Motors in Sequence

In this test, the faults are injected into motors 4 and 2 at 48 and 54.5 s, respectively. The hexacopter is made to hover at the height of 1.35 m. As can be seen in Figure 11, the attitude angles can track the desired angles although the they have some small oscillations after faults occur in motors 4 and 2. The residuals of system are presented in Figure 12. Unlike the case in Section 5.2.2, after the fault occurs in motor 4, the system can still detect and isolate the fault of motor 2.
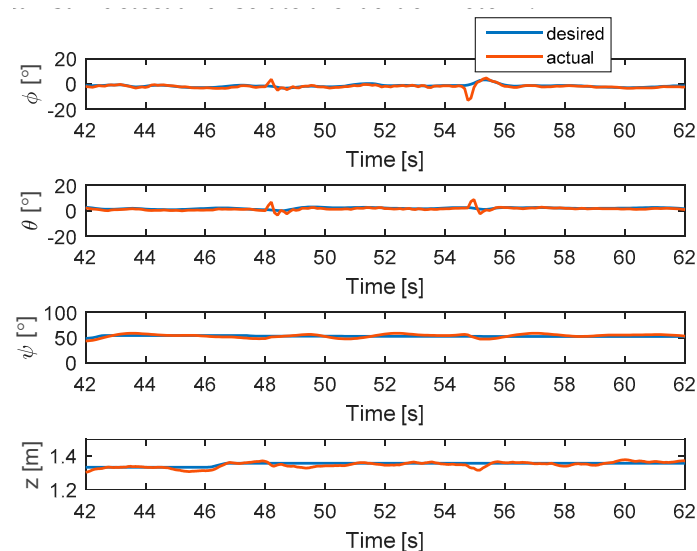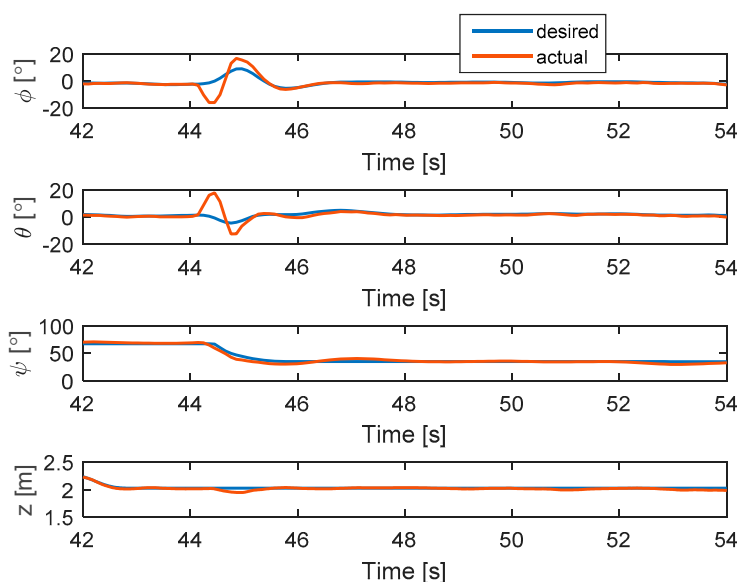

Figure 11. States of the system.

Figure 12. Residuals generated from fault detection schemes.

Figure 13 shows the detection unit of the system. It is clear that the faults in motors 4 and 2 are detected from the FD unit because their residuals' magnitudes exceed the threshold values. The corresponding output signals are presented in Figure 14. It shows that the PWMs of motors 4 and 2 have minimum values due to fault injections. The other PWMs are changed correspondingly to maintain the position of the hexacopter because the motors 4 and 2 are isolated from the allocation matrix.

Figure 13. Fault detection of actuators 2 and 4.
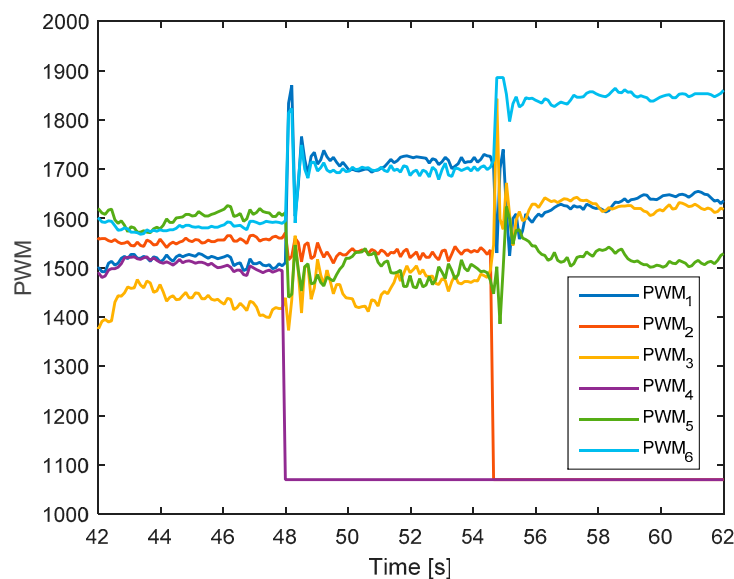
Figure 14. Output signals of the system.

The experimental work can be found at https://www.youtube.com/watch?v=yv-L4qBZ8EU or in the Supplementary Materials.

5.2.4. Complete Loss in Two Motors at the Same Time

In this scenario, the faults are injected to motors 2 and 4 at the same time. The hexacopter is made to hover at the height of 2 m and the fault occur at 44 s. As can be seen in Figure 15, the attitude angles can track the desired angles although the they have some oscillations after faults occur in motors 4 and 2. The residuals of system are presented in Figure 16. It is shown that when a fault occurs, all signs of the residuals are similar to those in Table 1 at the beginning and then they converge to zero. This convergence is because the residuals exceed the threshold values and two motor failures are isolated.
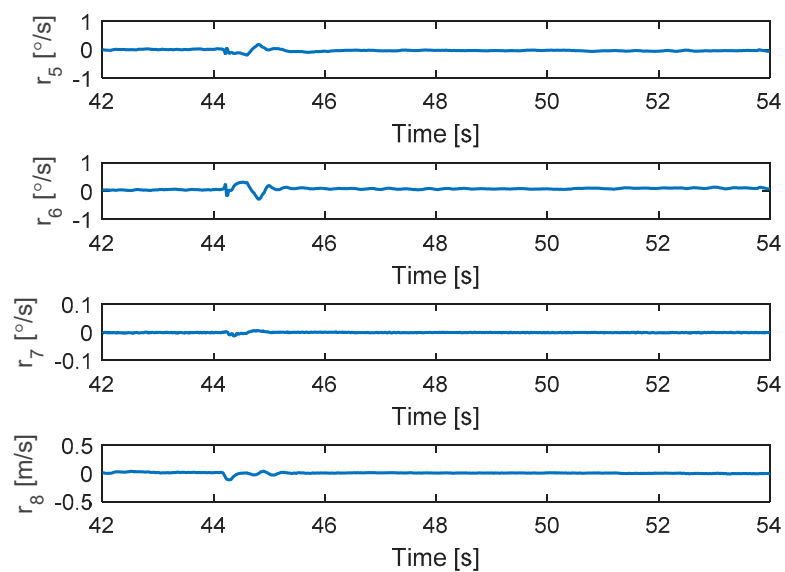

Figure 15. States of the system.

Figure 16. Residuals generated from fault detection schemes.

Figure 17 shows the detection unit of system. It is clear that the faults in motors 4 and 2 are detected from the FD unit at the same time because their residuals' magnitudes exceed the threshold values. Figure 18 shows the corresponding output signals. It shows that the PWMs of motors 4 and 2 have minimum values due to fault injections. The other PWMs are changed correspondingly to maintain the position of the hexacopter because motors 4 and 2 are isolated from the allocation matrix.
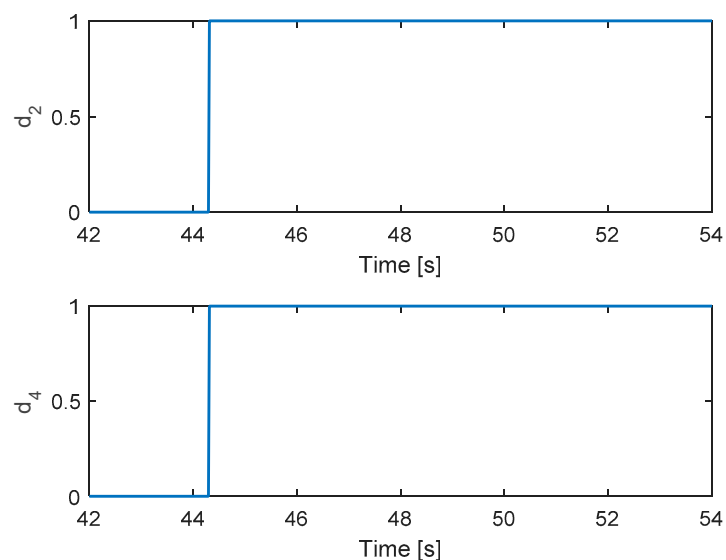
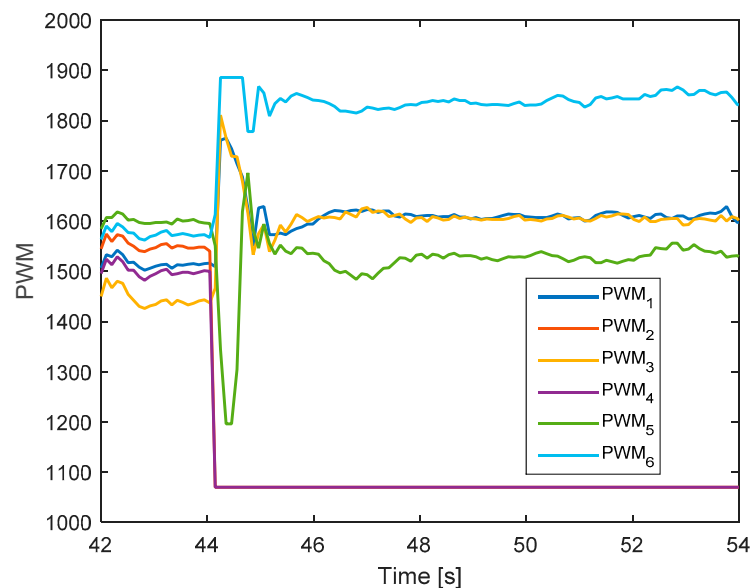**Figure 17.** Fault detection of actuators 2 and 4.

Figure 18. Output signals of the system.

The experimental work can be found at https://www.youtube.com/watch?v=uZuG44NqL8s or in the Supplementary Materials.

Remark 3. *In this research, two motor failures are presented. It should be mentioned that the total thrust generated from remaining motors should be larger than the gravity force of the hexacopter. Therefore, users should choose suitable motors and hexacopter masses for safety purposes.*

Remark 4. *Some previous studies assume that the location of fault is known, and they only focus on fault-tolerant control design to handle fault. In comparison with previous studies, our research can maintain the hexacopter up to two motor failures in presence of disturbances with two scenarios: sequential faults and simultaneous faults. We believe that our research could be innovative in commercial software of the Pixhawk2 flight controller.*

Remark 5. *In several flight tests on the Pixhawk2 flight controller, the hexacopter crashed because it lacks the fault detection and fault-tolerant control algorithm. Therefore, it is difficult to gather data of the Pixhawk2 flight controller for comparison.*

Remark 6. *Consider the frames that have more than six motors, the fault detection and fault-tolerant control schemes are more complex, which is out of scope of this research and will be investigated in further research.*

6. Conclusions

In this paper, the dynamic model of a hexacopter was explored with regard to disturbances and actuator faults. A fault detection based on a Thau observer was applied to monitor the location of one or two rotor failures. When the residuals exceeded the threshold values, the motor failure was isolated and then allocation matrix was changed. The results show that the proposed fault detection and isolation can be used to reconfigure the geometry of a hexacopter, hence making a safe flight when a fault occurs. Compared with previous studies, this research is innovative in several ways. First, the fault detection is integrated into fault-tolerant control as a single unit. Second, with the proposed disturbance observer-based sliding mode control, the discontinuous control gain is reduced, and it is required to be larger than the bound of estimation error rather than that of disturbance. Finally, the experimental works show that the hexacopter can handle complete fault in two cases: sequential fault and simultaneous fault. However, this study may not cover all cases of one motor or two motor

failures. This limitation can be addressed by rotation of the coordinates or scarifying the yaw control. Future work should consider this problem and propose a suitable method for implementation.

References

1. Zhao, W.; Go, T.H. Quadcopter formation flight control combining MPC and robust feedback linearization. *J. Frankl. Inst.* **2014**, *351*, 1335–1355. [CrossRef]
2. Mahmood, A.; Kim, Y. Decentralized formation flight control of quadcopters using robust feedback linearization. *J. Frankl. Inst.* **2017**, *354*, 852–871. [CrossRef]
3. Yang, S.; Ying, J.; Lu, Y.; Li, Z. Precise quadrotor autonomous landing with SRUKF vision perception. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
4. Shakernia, O.; Ma, Y.; Koo, T.J.; Sastry, S. Landing an unmanned air vehicle: vision based motion estimation and nonlinear control. *Asian J. Control* **1999**, *1*, 128–145. [CrossRef]
5. Ren, W.; Beard, R.W. Trajectory tracking for unmanned air vehicles with velocity and heading rate constraint. *IEEE Trans. Control Syst. Technol.* **2004**, *12*, 12–706. [CrossRef]
6. Bonna, R.; Camino, J.F. Trajectory Tracking Control of a Quadcopter Using Feedback Linearization. In Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics, Natal- Rio Grande Do Norte, Brazil, 22–27 February 2015.
7. Al-Naji, A.; Perera, A.G.; Mohammed, S.L.; Chahl, J. Life Signs Detector Using a Drone in Disaster Zones. *Remote Sens.* **2019**, *11*, 2441. [CrossRef]
8. Lippitt, C.D.; Zhang, S. The impact of small unmanned airborne platforms on passive optical remote sensing: A conceptual perspective. *Int. J. Remote Sens.* **2018**, *39*, 4852–4868. [CrossRef]
9. Burdziakowski, P. UAV in Todays Photogrammetry-Application Areas and Challenges. In Proceedings of the 18th International Multidisciplinary Scientific GeoConference SGEM, Albena, Bulgaria, 24 August–2 September 2018.
10. Burdziakowski, P. UAV Design and Construction for Real Time Photogrammetry and Visual Navigation. In Proceedings of the Baltic Geodetic Congress (BGC Geomatics), Olsztyn, Poland, 21–23 June 2018.
11. Lanzon, A.; Freddi, A.; Longhi, S. Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure. *J. Guid. Control Dyn.* **2014**, *37*, 580–591. [CrossRef]
12. Kacimi, A.; Mokhtari, A.; Kouadri, B. Sliding mode control based on adaptive backstepping approach for quadrotor unmanned aerial vehicle. *Przegląd Elektrotechniczny* **2012**, *88*, 188–193.
13. Sharifi, F.; Mirzaei, M.; Gordon, B.W.; Zhang, Y.M. Fault tolerant control of a quadrotor UAV using sliding mode control. In Proceedings of the Conference on Control and Fault Tolerant Systems, Nice, France, 6–7 October 2010; pp. 239–244.
14. Hu, Q.; Xiao, B. Adaptive fault tolerant control using integral sliding mode strategy with application to flexible spacecraft. *Int. J. Syst. Sci.* **2013**, *44*, 2273–2286. [CrossRef]
15. Barghandan, S.; Badamchizadeh, M.A.; Jahed-Motlagh, M.R. Improve Adaptive Fuzzy Sliding Mode Controller for Robust Fault Tolerant of a Quadrotor. *Int. J. Control Autom. Syst.* **2017**, *15*, 427–441. [CrossRef]
16. Yang, H.; Jiang, B.; Zhang, K. Direct self-repairing control of the quadrotor helicopter based on adaptive sliding mode control technique. In Proceedings of the 2014 IEEE Chinese Guidance, Navigation and Control Conference, Yantai, China, 8–10 August 2014.

17. He, J.; Qi, R.; Jiang, B.; Qian, J. Adaptive output feedback fault tolerant control design for hypersonic flight vehicles. *J. Frankl. Inst.* **2015**, *352*, 1811–1835. [CrossRef]

18. Merheb, A.-R.; Noura, H.; Bateman, F. Emergency control of AR Drone Quadrotor UAV suffering a total loss of one rotor. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 961–971. [CrossRef]

19. Lippiello, V.; Ruggiero, F.; Serra, D. Emergency landing for a quadrotor in case of a propeller failure: A backstepping approach. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014.

20. Mueller, M.W.; Andrea, R.D. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014.

21. Schneider, T.; Ducard, G.; Konrad, R.; Pascal, S. Fault-tolerant Control Allocation for Multirotor Helicopters Using Parametric Programming. In Proceedings of the International Micro Air Vehicle Conference and Flight Competition (IMAV), Braunschweig, Germany, 3–6 July 2012.

22. Falconi, G.P.; Marvakov, V.A.; Holzapfel, F. Fault Tolerant Control for a hexarotor system using incremental backstepping. In Proceedings of the IEEE Conference on Control Applications (CCA), Buenos Aires, Argentina, 19–22 September 2016.

23. Lee, J.; Choi, H.S.; Shim, H. Fault Tolerant Control of Hexacopter for Actuator Faults using Time Delay Control Method. *Int. J. Aeronaut. Space Sci.* **2016**, *17*, 54–63. [CrossRef]

24. Mazeh, H.; Saied, M.; Shraim, H.; Francis, C. Fault-tolerant control of an hexarotor unmanned aerial vehicle applying outdoor tests and experiments. *Int. Fed. Autom. Control* **2018**, *51*, 312–317. [CrossRef]

25. Dong, W.; Gu, G.-Y.; Zhu, X.; Ding, H. Modeling and Control of a Quadrotor UAV with Aerodynamics Concepts. *World Acad. Sci. Eng. Technol.* **2013**, *7*, 901–906.

26. Nguyen, N.P.; Hong, S.K. Fault Diagnosis and Fault-Tolerant Control Scheme for Quadcopter UAVs with a Total Loss of Actuator. *Energies* **2019**, *12*, 1139. [CrossRef]

27. Chen, W.-H.; Balance, D.J.; Gawthrop, P.J.; O'Reilly, J. A nonlinear disturbance observer for robotic manipulators. *IEEE Trans. Ind. Electron.* **2000**, *47*, 932–938. [CrossRef]

28. Cen, Z.; Noura, H.; Susilo, T.B.; Younes, Y.A. Robust fault diagnosis for quadrotor UAVs using adaptive Thau observer. *J. Intell. Robot. Syst.* **2014**, *73*, 573–588. [CrossRef]

29. DJI F550 Frame. Available online: https://www.dji.com/kr/flame-wheel-arf/feature (accessed on 22 October 2019).

30. Pixhawk2. Available online: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk-2.html (accessed on 22 October 2019).

31. PWM. Available online: http://ardupilot.org/dev/docs/learning-ardupilot-rc-input-output.html (accessed on 22 October 2019).

32. Building C++ Program from Eclipse Software. Available online: http://ardupilot.org/dev/docs/editing-the-code-with-eclipse.html (accessed on 7 October 2019).

33. Mission Planner. Available online: http://ardupilot.org/planner (accessed on 22 October 2019).

34. Telemetry. Available online: http://ardupilot.org/copter/docs/common-telemetry-landingpage.html (accessed on 22 October 2019).